CLAIMS

What is claimed is:

1. A method of maintaining version compatibility between a first computer program module and one or more interacting computer program modules that interact with the first module through an interface, wherein the modules are stored in computer storage, the method comprising the computer-implemented steps of:

creating and storing first information describing the interface at a plurality of times with a corresponding plurality of instances of a data structure;

creating and storing a mapping that associates the plurality of instances with a corresponding plurality of version numbers for the first module;

automatically developing a second version number for a second module of the one or more interacting modules based on the plurality of instances of the data structure and the mapping;

determining compatibility of the modules based on a first version number for the first module and the second version number for the second module.

2. A method as recited in Claim 1, wherein the step of automatically developing comprises:

describing a subset of the interface, which subset is employed by the second module;

determining of the plurality of instances at least one instance including data describing the subset of the interface; and

assigning the second version number for the second module based on the mapping and the at least one instance.

3. A method as recited in Claim 2, wherein said step of assigning the second version number comprises assigning as the second version number a particular value of the plurality of version numbers for the first module, the particular value associated with a particular instance of the at least one instance, the particular instance corresponding to an earliest time of one or more times corresponding to the at least one instance.

4. A method as recited in Claim 1, wherein the plurality of version numbers for the first module corresponding to the plurality of instances vary in one direction with time of the plurality of times corresponding to the plurality of instances.

5. A method as recited in Claim 1, wherein said step of describing the interface includes generating and storing in a first instance of the data structure data indicating signatures of a plurality of routines of the interface at a first time, wherein a signature of each routine includes a name of the routine and a type of the routine and parameter types for all parameters of the routine.

6. A method as recited in Claim 1, wherein:

each of the interacting computer program modules include instructions causing one or more processors to obtain at least one of a property of a corresponding networking device type of a plurality of networking devices types and an action performed by the corresponding networking device type;

the first computer program module includes instructions causing one or more processors, based on interacting with a particular interacting computer program module, to perform at least one of communicating with a first device of the corresponding networking device type on a network of networking devices including the first device, and presenting properties of the first device to a network manager for the network, and displaying connections among the networking devices of the network to the network manager; and

the plurality of networking devices types include one or more models of repeater, a switch, a router, a hub, a bridge, and a gateway.

7. A method as recited in Claim 1, wherein the second version number is developed when the second module is developed; and compatibility is determined at a later time.

8. A method of maintaining version compatibility between a first module and one or more interacting modules that interact with the first module through an interface, the method comprising the steps of:

retrieving data from a stored mapping between a plurality of instances of a data structure describing the interface at a corresponding plurality of times and a corresponding plurality of version numbers for the first module;

automatically developing a second version number for a second module of the one or more interacting modules based on the mapping;

determining compatibility based on a first version number for the first module and the second version number for the second module.

9. A method as recited in Claim 8, said step of developing the second version number further comprising the steps of:

describing a subset of the interface, which subset is employed by the second module;

determining of the plurality of instances at least one instance including data describing the subset of the interface; and

assigning the second version number for the second module based on the mapping and the at least one instance.

10. A method as recited in Claim 9, wherein said step of assigning the second version number comprises assigning as the second version number a particular value of the plurality of version numbers for the first module, the particular value associated with a particular instance of the at least one instance, the particular instance corresponding to an earliest time of one or more times corresponding to the at least one instance.

11. A method as recited in Claim 8, wherein the plurality of version numbers for the first module corresponding to the plurality of instances vary in one direction with time of the plurality of times corresponding to the plurality of instances.

12. A method as recited in Claim 8, wherein a first instance of the data structure comprises data indicating a routine name and a routine type of a routine of the first module at a first time.

13. A method as recited in Claim 12, wherein the first data further indicates a parameter type for the routine.

14. A method as recited in Claim 8, wherein a first instance of the data structure comprises data indicating signatures of a plurality of routines of the interface at a first time, wherein a signature of each routine includes a name of the routine and a type of the routine and parameter types for all parameters of the routine.

15. A method as recited in Claim 14, wherein the data indicating signatures of the plurality of routines of the interface at a first time comprises hashed values, each hashed value uniquely indicating a signature of each routine of the interface.

16. A method as recited in Claim 14, wherein the plurality of routines comprises all the routines of the interface.

17. A method as recited in Claim 14, wherein the plurality of routines comprises all the routines of the interface except routines not implemented in the first module.

18. A method as recited in Claim 8, wherein:
the second version number is developed when the second module is developed; and
compatibility is determined at a later time.

19. A method as recited in Claim 18, wherein compatibility is determined when the second module is installed for use with the first module.

20. A method as recited in Claim 18, wherein compatibility is determined when the second module is invoked for execution by the first module.

21. A method as recited in Claim 8, wherein each module of the first module and the one or more interacting modules comprises instructions for causing one or more processors to perform one or more tasks.

22. A method as recited in Claim 8, wherein the first module comprises instructions for causing one or more processors to manage a plurality of networking devicess in response to data indicating input by a user.

23. A method as recited in Claim 8, wherein each interacting module of the one or more interacting modules comprises instructions for causing one or more processors to provide device-specific information for one of a plurality of networking devicess.

24. A method as recited in Claim 21, wherein:

each of the interacting modules include instructions causing one or more processors to obtain at least one of a property of a corresponding networking device type of a plurality of networking devices types and an action performed by the corresponding networking device type;

the first module includes instructions causing one or more processors, based on interacting with a particular interacting computer program module, to perform at least one of communicating with a first device of the corresponding networking device type on a network of networking devices including the first device, and presenting properties of the first device to a network manager for the network, and displaying connections among the networking devices of the network to the network manager; and

the plurality of networking devices types include one or more models of repeater, a switch, a router, a hub, a bridge, and a gateway.

25. A method of determining version compatibility between a first module and a second module of one or more interacting modules that interact with the first module through an interface, the method comprising the steps of:

obtaining a first version number for the first module;

obtaining a second version number for the second module, the second version number set when the second module is developed based on a mapping between a plurality of instances of a data structure describing the interface at a corresponding plurality of times and a corresponding plurality of version numbers for the first module; and

determining whether the modules are compatible based on the first version number and the second version number.

26. A method as recited in Claim 25, wherein:

each of the interacting modules include instructions causing one or more processors to obtain at least one of a property of a corresponding networking device type of a plurality of networking devices types and an action performed by the corresponding networking device type;

the first module includes instructions causing one or more processors, based on interacting with a particular interacting computer program module, to perform at least one of communicating with a first device of the corresponding networking device type on a network of networking devices including the first device, and presenting properties of the first device to a network manager for the network, and displaying connections among the networking devices of the network to the network manager; and

the plurality of networking devices types include one or more models of repeater, a switch, a router, a hub, a bridge, and a gateway.

27. A computer-readable medium for maintaining version compatibility between a first module and one or more interacting modules that interact with the first module through an interface, the computer-readable medium carrying:

a plurality of instances of a data structure describing the interface at a corresponding plurality of times;

a mapping that associates the plurality of instances with a corresponding plurality of version numbers for the first module; and

one or more sequences of instructions, which, when executed by one or more processors, cause the one or more processors to carry out the steps of

retrieving data from the mapping, and

developing a second version number for a second module of the one or more interacting modules,

wherein compatibility is determined based on a first version number for the first module and the second version number for the second module.

28. A system for maintaining version compatibility between a first module and one or more interacting modules that interact with the first module through an interface, the system comprising:

means for retrieving a stored mapping between a plurality of instances of a data structure describing the interface at a corresponding plurality of times and a corresponding plurality of version numbers for the first module;

means for automatically developing a second version number for a second module of the one or more interacting modules based on the mapping; and

means for determining compatibility based on a first version number for the first module and the second version number for the second module.

29. A computer system for maintaining version compatibility between a first module and one or more interacting modules that interact with the first module through an interface, the system comprising:

a processor;

a computer-readable medium carrying

a stored mapping between a plurality of instances of a data structure describing the interface at a corresponding plurality of times and a corresponding plurality of version numbers for the first module, and

one or more stored sequences of instructions which, when executed by the processor, cause the processor to carry out the steps of:

retrieving data from the mapping;

developing a second version number for a second module of the one or more interacting modules based on the mapping; and

determining compatibility based on a first version number for the first module and the second version number for the second module.